

On the Joint Decoding of LDPC Codes and Finite-State Channels via Linear Programming

Byung-Hak Kim and Henry D. Pfister

Department of Electrical and Computer Engineering, Texas A&M University

Email: {bhkim,hpfister}@tamu.edu

Abstract—In this paper, the linear programming (LP) decoder for binary linear codes, introduced by Feldman, et al. is extended to joint-decoding of binary-input finite-state channels. In particular, we provide a rigorous definition of LP joint-decoding pseudo-codewords (JD-PCWs) that enables evaluation of the pairwise error probability between codewords and JD-PCWs. This leads naturally to a provable upper bound on decoder failure probability. If the channel is a finite-state intersymbol interference channel, then the LP joint decoder also has the maximum-likelihood (ML) certificate property and all integer valued solutions are codewords. In this case, the performance loss relative to ML decoding can be explained completely by fractional valued JD-PCWs.

I. INTRODUCTION

A. Motivation

Message-passing iterative decoding has been a very popular decoding algorithm in research and practice for the past fifteen years [1]. In the last five years, linear programming (LP) decoding has been a popular topic in coding theory and has given new insight into the analysis of iterative decoding algorithms and their modes of failure [2][3][4]. For both decoders, fractional vectors, known as pseudo-codewords (PCWs), play an important role in the performance characterization of these decoders [3][5]. This is in contrast to classical coding theory where the performance of most decoding algorithms (e.g., maximum-likelihood (ML) decoding) is completely characterized by the set of codewords.

For channels with memory, such as finite-state channels (FSCs), the situation is a bit more complicated. In the past, one typically separated channel decoding (i.e., estimating the channel inputs from the channel outputs) from error-correcting code (ECC) decoding (i.e., estimating the transmitted codeword from estimates of the channel inputs) [6]. The advent of message-passing iterative decoding enabled the joint-decoding (JD) of the channel and code by iterating between these two decoders [7].

In this paper, we extend the LP decoder to the JD of binary-input FSCs and define LP joint-decoding pseudo-codewords (JD-PCWs). This leads naturally to a provable upper bound (e.g., a union bound) on the probability of decoder failure as a sum over all codewords and JD-PCWs. This extension has been considered as a challenging open problem in the prior

work [8][2]. The problem is well posed by Feldman in his PhD thesis [2, Section 9.5 page 146],

"In practice, channels are generally not memoryless due to physical effects in the communication channel." ... "Even coming up with a proper linear cost function for an LP to use in these channels is an interesting question. The notions of pseudocodeword and fractional distance would also need to be reconsidered for this setting."

Other than providing satisfying answer to the above open question, our primary motivation is the prediction of the error rate for joint decoding at high SNR. The idea is to run a simulation at low SNR and keep track of all observed codeword and pseudo-codeword errors. A truncated union bound is computed by summing over all observed errors and the result is an estimate of the error rate at high SNR. Computing this bound is complicated by the fact that the loss of channel symmetry implies that the dominant PCWs may depend on the transmitted sequence.

While we were preparing this manuscript, we became aware of a more general approach by Flanagan [9][10]. In fact, our LP formulation was developed independently but is identical to his "Efficient LP relaxation". Our motivation, however, is somewhat different. The main goal is to use the error rate of joint LP decoding as a tool to analyze joint iterative decoding of FSCs and low-density parity-check (LDPC) codes. Thus, we give novel prediction results in Sec. V. We also observe that both formulations provide an ML edge-path certificate that is not equivalent to an ML codeword certificate (see Remark 1 and 2). This property is not guaranteed by Wadayama's approach based on quadratic programming [8].

The paper is structured as follows. After briefly reviewing LP decoding and FSCs in the remainder of Sec. I, we describe the LP joint decoder in Sec. II and define JD-PCWs in Sec. III. In Sec. IV, we discuss the decoder performance analysis via the union bound (and pairwise error probability) over JD-PCWs and notions of generalized Euclidean distance. Experimental results are given in Sec. V and conclusions are given in Sec. VI.

B. Background

Feldman, et al. introduced the LP decoding for binary linear codes in [3][2]. It is based on solving an LP relaxation of an integer program which is equivalent to ML decoding. Later this method was extended to codes

over larger alphabets [11] and to the simplified decoding of intersymbol interference (ISI) [12]. For long codes, the performance of LP decoding is slightly inferior to iterative decoding but, unlike the iterative decoder, the LP decoder either detects a failure or outputs a codeword which is guaranteed to be the ML codeword.

Let $\mathcal{C} \subseteq \{0, 1\}^n$ be the length- n binary linear code defined by the parity-check matrix H and $\mathbf{c} = (c_1, \dots, c_n)$ be a codeword. If \mathcal{I} is the set whose elements are the sets of indices involved in each parity check, then we have

$$\mathcal{C} = \left\{ \mathbf{c} \in \{0, 1\}^n \mid \sum_{i \in I} c_i \equiv 0 \pmod{2}, \forall I \in \mathcal{I} \right\}.$$

The *codeword polytope* is the convex hull of \mathcal{C} . This polytope can be quite complicated to describe though, so instead one constructs a simpler polytope using local constraints. Each parity-check $I \in \mathcal{I}$ defines a local constraint that can also be viewed as a polytope in $[0, 1]^n$.

Definition 1: The *local codeword polytope* $\text{LCP}(I)$ associated with a parity check is the convex hull of the bit sequences that satisfy the check. It is given explicitly by

$$\text{LCP}(I) \triangleq \bigcap_{\substack{S \subseteq I \\ |S| \text{ odd}}} \left\{ \mathbf{c} \in [0, 1]^n \mid \sum_{i \in S} c_i - \sum_{i \in I-S} c_i \leq |S| - 1 \right\}.$$

Definition 2: The *relaxed polytope* $\mathcal{P}(H)$ is the intersection of the LCPs over all checks, so

$$\mathcal{P}(H) \triangleq \bigcap_{I \in \mathcal{I}} \text{LCP}(I).$$

Theorem 1 ([2]): Consider n consecutive uses of a symmetric channel $\Pr(Y = y | C = c)$. If a uniform random codeword is transmitted and $\mathbf{y} = (y_1, \dots, y_n)$ is received, then the LP decoder outputs $\mathbf{f} = (f_1, \dots, f_n)$ given by

$$\arg \min_{\mathbf{f} \in \mathcal{P}(H)} \sum_{i=1}^n f_i \log \left(\frac{\Pr(Y_i = y_i | C_i = 0)}{\Pr(Y_i = y_i | C_i = 1)} \right),$$

which is the ML solution if \mathbf{f} is integral (i.e., $\mathbf{f} \in \{0, 1\}^n$).

Definition 3: An *LP decoding pseudo-codeword* (LPD-PCW) of a code defined by the parity-check matrix H is any vertex of the relaxed (fundamental) polytope $\mathcal{P}(H)$.

Definition 4: A *finite-state channel* (FSC) defines a probabilistic mapping from a sequence of inputs to a sequence of outputs. Each output $Y_i \in \mathcal{Y}$ depends only on the current input $X_i \in \mathcal{X}$ and channel state $S_i \in \mathcal{S}$ instead of the entire history of inputs and channel states. Mathematically, we have $P(y, s' | x, s) \triangleq \Pr(Y_i = y, S_{i+1} = s' | X_i = x, S_i = s)$ for all i , and we use the shorthand $P(y_1^n, s_2^{n+1} | x_1^n, s_1)$ for

$$\begin{aligned} \Pr(Y_1^n = y_1^n, S_2^{n+1} = s_2^{n+1} | X_1^n = x_1^n, S_1 = s_1) \\ = \prod_{i=1}^n P(y_i, s_{i+1} | x_i, s_i). \end{aligned}$$

Definition 5: A *finite-state intersymbol interference channel* (FSISIC) is a FSC whose next state is a deterministic

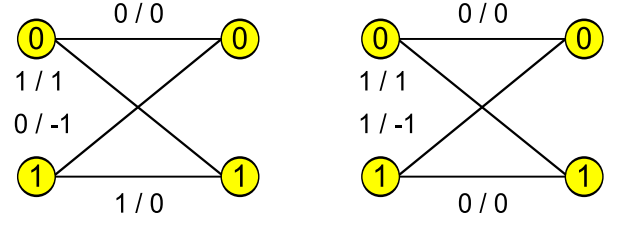


Figure 1. State diagrams for noiseless decode channel with and without precoding. The edges are labeled by the input/output pair. function, $\eta(x, s)$, of the current state s and input x . Mathematically, this implies that

$$\sum_{y \in \mathcal{Y}} P(y, s' | x, s) = \begin{cases} 1 & \text{if } \eta(x, s) = s' \\ 0 & \text{otherwise} \end{cases}.$$

Definition 6: The *decode channel* (DIC) is a binary-input FSISIC channel with a linear response of $G(z) = 1 - z^{-1}$ and Gaussian noise. If the input bits are differentially encoded prior to transmission, then the resulting channel is called the *precoded decode channel* (pDIC). The state diagrams of these two channels are shown in Fig. 1.

II. NEW RESULTS: LP JOINT-DECODING

Now, we describe the **LP joint decoder** in terms of the trellis of the FSC and the checks in the binary linear code. Let n be the length of the code and \mathbf{y} be the received sequence. The trellis consists of $(n+1)|\mathcal{S}|$ vertices (i.e., one for each state and time) and a set \mathcal{E} of at most $2n|\mathcal{S}|^2$ edges (not $2n|\mathcal{S}|$, i.e., one edge for each input-labeled state transition and time). For each edge $e \in \mathcal{E}$, the functions $t(e) \rightarrow \{1, \dots, n\}$, $s(e) \rightarrow \mathcal{S}$, $s'(e) \rightarrow \mathcal{S}$, $x(e) \rightarrow \{0, 1\}$, and $a(e) \rightarrow \mathcal{A}$ map this edge to its respective time index, initial state, final state, input bit, and noiseless output symbol. The LP formulation requires one variable for each edge $e \in \mathcal{E}$, and we denote that variable by $g(e)$. Likewise, the LP decoder requires one cost variable for each edge and we use the branch metric

$$b(e) \triangleq \begin{cases} -\ln P(y_{t(e)}, s'(e) | x(e), s(e)) & \text{if } t(e) > 1 \\ -\ln (P(y_{t(e)}, s'(e) | x(e), s(e)) P(s(e))) & \text{if } t(e) = 1 \end{cases}.$$

Definition 7: The *trellis polytope* \mathcal{T} enforces the flow conservation constraints for channel decoder. The flow constraint for state j at time i is given by

$$\mathcal{F}_{ij} \triangleq \left\{ g(\cdot) \in [0, 1]^{|\mathcal{E}|} \mid \sum_{\substack{e \in \mathcal{E}: \\ t(e)=i, \\ s'(e)=j}} g(e) = \sum_{\substack{e \in \mathcal{E}: \\ t(e)=i+1, \\ s(e)=j}} g(e) \right\}.$$

Using this, the *trellis polytope* \mathcal{T} is given by

$$\mathcal{T} \triangleq \left\{ g(\cdot) \in \bigcap_{\substack{i=1, \dots, n-1 \\ j \in \mathcal{S}}} \mathcal{F}_{ij} \mid \sum_{\substack{e \in \mathcal{E}: \\ t(e)=1}} g(e) = 1 \right\}.$$

Theorem 2 ([2]): Finding the ML edge-path through a weighted trellis is equivalent to solving the minimum-cost flow LP

$$\arg \min_{g(\cdot) \in \mathcal{T}} \sum_{e \in \mathcal{E}} g(e)b(e)$$

and the optimum $g(\cdot)$ must be integral (i.e., $g(\cdot) \in \{0, 1\}^{|\mathcal{E}|}$) unless there are ties.

Definition 8: Let \mathcal{Q} be the projection of $g(\cdot)$ onto the input vector $\mathbf{f} = (f_1, \dots, f_n) \in [0, 1]^n$ where $\mathbf{f} = \mathcal{Q}\mathbf{g}$ and

$$f_i = \sum_{e \in \mathcal{E}: t(e)=i, x(e)=1} g(e).$$

Definition 9: The trellis-wise relaxed polytope $\mathcal{P}_{\mathcal{T}}(H)$ for $\mathcal{P}(H)$ is defined by

$$\mathcal{P}_{\mathcal{T}}(H) \triangleq \{g(\cdot) \in \mathcal{T} | \mathcal{Q}g \in \mathcal{P}(H)\}.$$

Definition 10: The set of trellis-wise codewords $\mathcal{C}_{\mathcal{T}}$ for \mathcal{C} is defined as

$$\mathcal{C}_{\mathcal{T}} \triangleq \left\{ g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H) \mid g(\cdot) \in \{0, 1\}^{|\mathcal{E}|} \right\}.$$

Theorem 3: The LP joint decoder computes

$$\arg \min_{g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H)} \sum_{e \in \mathcal{E}} g(e)b(e)$$

and outputs a joint ML edge-path if $g(\cdot)$ is integral.

Proof: Let \mathcal{V} be the set of valid input/state sequence pairs. For a given \mathbf{y} , the ML edge-path decoder computes

$$\begin{aligned} & \arg \max_{(x_1^n, s_1^{n+1}) \in \mathcal{V}} P(y_1^n, s_2^{n+1} | x_1^n, s_1) P(s_1) \\ &= \arg \max_{g(\cdot) \in \mathcal{C}_{\mathcal{T}}} P(s_1) \prod_{e \in \mathcal{E}: g(e)=1} P(y_{t(e)}, s'(e) | x(e), s(e)) \\ &= \arg \min_{g(\cdot) \in \mathcal{C}_{\mathcal{T}}} \sum_{e \in \mathcal{E}: g(e)=1} b(e) \\ &= \arg \min_{g(\cdot) \in \mathcal{C}_{\mathcal{T}}} \sum_{e \in \mathcal{E}} g(e)b(e), \end{aligned}$$

where ties are resolved in a systematic manner and $b(e)$ at $t(e) = 1$ has an additional initial state term as $-\ln P(s(e))$. By relaxing $\mathcal{C}_{\mathcal{T}}$ into $\mathcal{P}_{\mathcal{T}}(H)$, we obtain the desired result. ■

Corollary 1: For a FSISIC, the LP joint decoder outputs a joint ML codeword if $g(\cdot)$ is integral.

Proof: The joint ML decoder for codewords computes

$$\begin{aligned} & \arg \max_{x_1^n \in \mathcal{C}} \sum_{s_2^{n+1} \in \mathcal{S}^n} P(y_1^n, s_2^{n+1} | x_1^n, s_1) P(s_1) \\ &= \arg \max_{x_1^n \in \mathcal{C}} \sum_{s_2^{n+1} \in \mathcal{S}^n} \prod_{i=1}^n P(y_i, s_{i+1} | x_i, s_i) P(s_1) \\ &\stackrel{(a)}{=} \arg \max_{x_1^n \in \mathcal{C}} \prod_{i=1}^n P(y_i, \eta(x_i, s_i) | x_i, s_i) P(s_1) \\ &\stackrel{(b)}{=} \arg \min_{g(\cdot) \in \mathcal{C}_{\mathcal{T}}} \sum_{e \in \mathcal{E}} g(e)b(e), \end{aligned}$$

where (a) follows from Defn. 5 and (b) holds because each input sequence defines a unique edge-path. Therefore, the LP joint-decoder outputs an ML codeword if $g(\cdot)$ is integral. ■

Remark 1: If the channel is not a FSISIC (e.g., finite-state fading channel), the integer valued solutions of the LP joint-decoder are ML edge-paths and not necessarily ML codewords. This occurs because the decoder is unable to sum to the probability of the multiple edge-paths associated with the same codeword (e.g., if multiple distinct edge-paths are associated with the same input labels).

III. JOINT-DECODING PSEUDO-CODEWORDS

Pseudo-codewords have been observed and given names by a number of authors [1][13][14], but the simplest general definition was provided by Feldman, et al. in the context of LP decoding of parity-check codes [3]. One nice property of the LP decoder is that it always returns an integer codeword or a fractional pseudo-codeword. Vontobel and Koetter have shown that a very similar set of pseudo-codewords also affect message-passing decoders, and that they are essentially fractional codewords that cannot be distinguished from codewords using only local constraints [5]. We define JD-PCW this section because of their primary importance in the characterization of code performance at very low error rates.

Definition 11: The output of the LP joint decoder is a trellis-wise (ML) codeword (TCW) if $g(e) \in \{0, 1\}$ for all $e \in \mathcal{E}$. Otherwise, if $g(e) \in (0, 1)$ for some $e \in \mathcal{E}$, then the solution is called a joint-decoding trellis-wise pseudo-codeword (JD-TPCW) and the decoder outputs “failure”.

Definition 12: Any TCW \mathbf{g} can be projected onto a (symbol-wise) codeword (SCW) $\mathbf{f} = \mathcal{Q}\mathbf{g}$. Likewise, any JD-TPCW \mathbf{g} can be projected onto a joint-decoding symbolwise pseudo-codeword (JD-SPCW) $\mathbf{f} = \mathcal{Q}\mathbf{g}$.

Remark 2: For FSISIC, the LP joint decoder has the ML certificate property; if the decoder outputs a SCW, then it is guaranteed to be the ML codeword (see Cor. 1).

Definition 13: Any TCW can be projected onto a symbol-wise signal-space codeword (SSCW) and any JD-TPCW \mathbf{g} can be projected onto a joint-decoding symbol-wise signal-space pseudo-codeword (JD-SSPCW) $\mathbf{p} = (p_1, \dots, p_n)$ by averaging the components with

$$p_i = \sum_{e \in \mathcal{E}: t(e)=i} g(e)a(e).$$

Example 1: Consider the single parity-check code SPC(3,2). Over precoded dicode channel (starts in zero state) with AWGN, this code has five joint-decoding pseudo-codewords. A simulation was performed for joint-decoding of the SPC(3,2) on the pDIC trellis and the set of JD-TPCW, by ordering the trellis edges appropriately, was found to be

$$\begin{aligned} & \{(0100; 00.5.5; 0.5.50), (.5.500; .500.5; 0100), \\ & (.5.500; 0.5.50; 0010), (1000; .5.500; 0.5.50), \\ & (.5.500; .5000; 0.5.50)\}. \end{aligned}$$

Using \mathcal{Q} to project them into $\mathcal{P}(H)$, we get the corresponding set of JD-SPCW

$$\{(1, .5, .5), (.5, .5, 1), (.5, .5, 0), (0, .5, .5), (.5, 0, .5)\}.$$

IV. UNION BOUND FOR LP JOINT-DECODING

Now that we have defined the relevant pseudo-codewords, we turn our attentions to the question of “how bad” a certain pseudo-codeword is, i.e., we want to quantify pairwise error probabilities. In fact, we will use the insights gained in the previous section to obtain a union bound on the decoder word error probability (as a tight approximation) to analyze the performance of the proposed LP-joint decoder. Toward this end, let’s consider the pairwise error event between a SSCW \mathbf{c} and a JD-SSPCW \mathbf{p} first.

Theorem 4: A necessary and sufficient condition for the pairwise decoding error between a SSCW \mathbf{c} and a JD-SSPCW \mathbf{p} is

$$\sum_{e \in \mathcal{E}} g(e)b(e) \leq \sum_{e \in \mathcal{E}} \tilde{g}(e)b(e),$$

where $g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H)$ and $\tilde{g}(\cdot) \in \mathcal{C}_{\mathcal{T}}$ are the LP variables for \mathbf{p} and \mathbf{c} respectively.

For the moment, let \mathbf{c} be the SSCW of FSISIC to an AWGN channel whose output sequence is $\mathbf{y} = \mathbf{c} + \mathbf{v}$, where $\mathbf{v} = (v_1, \dots, v_n)$ is an i.i.d. Gaussian sequence with mean 0 and variance σ^2 . We will show that each pairwise probability has a simple closed-form expression that depends only on a generalized squared Euclidean distance $d_{gen}^2(\mathbf{c}, \mathbf{p})$ and the noise variance σ^2 . The next few definitions and theorems can be seen as a generalization of [15] and a special case of the more general formulation in [10].

Theorem 5: Let \mathbf{y} be the output of a FSISIC with zero-mean AWGN whose variance is σ^2 per output. Then, the LP joint decoder is equivalent to

$$\arg \min_{g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H)} \sum_{e \in \mathcal{E}} g(e) (y_{t(e)} - a(e))^2.$$

Proof: For each edge $e \in \mathcal{E}$, the output $y_{t(e)}$ is Gaussian with mean $a(e)$ and variance σ^2 , so we have $P(y_{t(e)}, s'(e) | x(e), s(e)) \sim \mathcal{N}(a(e), \sigma^2)$. Therefore, the LP joint-decoder computes

$$\arg \min_{g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H)} \sum_{e \in \mathcal{E}} g(e)b(e) = \arg \min_{g(\cdot) \in \mathcal{P}_{\mathcal{T}}(H)} \sum_{e \in \mathcal{E}} g(e) (y_{t(e)} - a(e))^2.$$

Definition 14: Let \mathbf{c} be a SSCW and \mathbf{p} a JD-SSPCW. Then the *generalized squared Euclidean distance* between \mathbf{c} and \mathbf{p} can be defined in terms of their trellis-wise descriptions by

$$d_{gen}^2(\mathbf{c}, \mathbf{p}) \triangleq \frac{(\|\mathbf{d}\|^2 + \sigma_p^2)^2}{\|\mathbf{d}\|^2}$$

where

$$\|\mathbf{d}\|^2 \triangleq \sum_{i=1}^n (c_i - p_i)^2, \quad \sigma_p^2 \triangleq \sum_{e \in \mathcal{E}} g(e)a^2(e) - \sum_{i=1}^n p_i^2.$$

Theorem 6: The pairwise error probability between a SSCW \mathbf{c} and a JD-SSPCW \mathbf{p} is

$$\Pr(\mathbf{c} \rightarrow \mathbf{p}) = Q\left(\frac{d_{gen}(\mathbf{c}, \mathbf{p})}{2\sigma}\right).$$

Proof: The pairwise error probability $\Pr(\mathbf{c} \rightarrow \mathbf{p})$ that the LP joint-decoder will choose the pseudo-codeword \mathbf{p} over \mathbf{c} can be written as $\Pr(\mathbf{c} \rightarrow \mathbf{p}) =$

$$\begin{aligned} & \Pr \left\{ \sum_{i=1}^n \sum_{\substack{e \in \mathcal{E}: \\ t(e)=i}} g(e) (y_{t(e)} - a(e))^2 \leq \sum_{i=1}^n (y_i - c_i)^2 \right\} \\ &= \Pr \left\{ \sum_i y_i (c_i - p_i) \leq \frac{1}{2} (\sum_i c_i^2 - \sum_{e \in \mathcal{E}} g(e)a^2(e)) \right\} \\ &\stackrel{(a)}{=} Q \left(\frac{\sum_i c_i (c_i - p_i) - \frac{1}{2} (\sum_i c_i^2 - \sum_{e \in \mathcal{E}} g(e)a^2(e))}{\sigma \sqrt{\sum_i (c_i - p_i)^2}} \right) \\ &\stackrel{(b)}{=} Q \left(\frac{\|\mathbf{d}\|^2 + \sigma_p^2}{2\sigma \|\mathbf{d}\|} \right) \stackrel{(c)}{=} Q \left(\frac{d_{gen}(\mathbf{c}, \mathbf{p})}{2\sigma} \right), \end{aligned}$$

where (a) follows from the fact that $\sum_i y_i (c_i - p_i)$ is distributed $\mathcal{N}(\sum_i c_i (c_i - p_i), \sum_i (c_i - p_i)^2)$ and (b)/(c) follow from Defn. 14. ■

Wiberg was the first to define a generalization of the Euclidean distance to explain errors caused by iterative decoding [1] and this was extended to non-binary cases [15] where $\Pr(\mathbf{c} \rightarrow \mathbf{p})$ looks very similar to Thm. 6. The main difference is the definition of the trellis-wise approach used for JD-TPCW.

The performance degradation of LP decoding relative to ML decoding can be explained by pseudo-codewords and their contribution to the error rate depends on $d_{gen}(\mathbf{c}, \mathbf{p})$. Indeed, by defining $K_{d_{gen}}(\mathbf{c})$ as the number of codewords and JD-PCWs at distance d_{gen} from \mathbf{c} and $\mathcal{G}(\mathbf{c})$ as the set of generalized Euclidean distances, we can write the union bound on word error rate (WER) as

$$P_{w|\mathbf{c}} \leq \sum_{d_{gen} \in \mathcal{G}(\mathbf{c})} K_{d_{gen}}(\mathbf{c}) Q\left(\frac{d_{gen}}{2\sigma}\right).$$

Of course, we need the set of JD-TPCWs to compute $\Pr(\mathbf{c} \rightarrow \mathbf{p})$ with the Thm. 6. There are two complications with this approach. One is that like original problem [2], no method is known yet for computing the generalized Euclidean distance spectrum, apart from going through all error events explicitly. Another is, unlike original problem, the constraint polytope may not be symmetric under codeword exchange. Therefore the decoder performance may not be symmetric under codeword exchange. Hence, the decoder performance may depend on the transmitted codeword. In this case, the pseudo-codewords will also depend on the transmitted sequence.

V. SIMULATION RESULTS AND ERROR RATE PREDICTION

In this section, we present simulation results for two LDPC codes on the precoded dicode channel (pDIC) and use those results to predict the error rate well beyond the limits of our simulations. Both codes are (3, 5)-regular binary LDPC codes; the first has length 155 and the second has length 455. The parity-check matrices were chosen randomly except that double-edges and four-cycles were avoided. Since the performance depends on the transmitted codeword, the results were obtained for 3 randomly chosen codewords of fixed

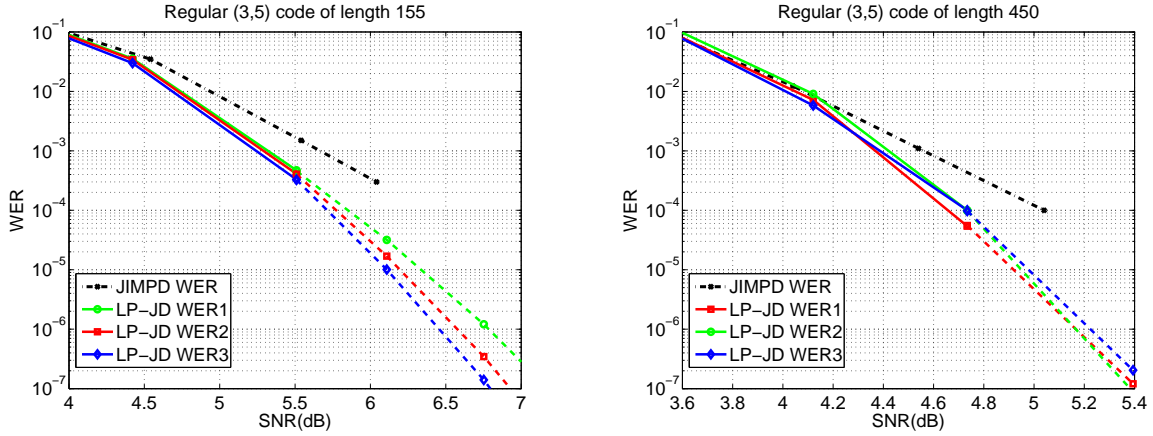


Figure 2. This figure shows comparison between the LP joint decoding and joint iterative message-passing decoding on the precoded dicode channel with AWGN for random (3,5) regular LDPC codes of length $n = 155$ (left) and $n = 450$ (right). The curves shown are the LP-JD WER (solid), LP-JD WER prediction (dashed) and JIMPD WER (dash-dot). The experiments were repeated for three different non-zero codewords in each case. The dashed curves are computed using the union bound in Sec. IV based on JD-PCWs observed at 3.4558 dB (left) 2.6696 dB (right) and the dash-dot curves are obtained using the state-based joint iterative message-passing decoder (JIMPD) described in [16]. Note that SNR is defined as channel output power divided by σ^2 .

weight. The weight was chosen to be roughly half the block length, giving weight 74 in the first case and 226 in the second case.

The results are shown in Fig. 2. The solid lines represent the simulation curves while the dashed lines represent a truncated union bound. The truncated union bound is obtained by computing the generalized Euclidean distances associated with all decoding errors that occurred at some low SNR points (e.g., WER of roughly than 10^{-1}) until we observe a stationary generalized Euclidean distance spectrum. This high WER allows the decoder to rapidly discover JD-PCWs. The dash-dot curves show the state-based joint iterative message-passing decoder (JIMPD) algorithm described in [16]. Somewhat surprisingly, we find that LP joint-decoding outperforms JIMPD by about 0.5dB at WER of 10^{-4} .

The LP decoding is performed in the dual domain because this is much faster than the primal when using MATLAB. Due to the slow speed of LP decoding still, simulations were completed up to a WER of roughly 10^{-5} . It is well-known that the truncated bound should be relatively tight at high SNR if all the dominant JD-PCWs have been found.

The final complication that must be discussed is the dependence on the transmitted codeword. It is known that long LDPC codes with joint iterative decoding experience a concentration phenomenon [16] whereby the error probability associated with transmitting a randomly chosen codeword is very close, with high probability, to the average error probability over all transmitted codewords. We note that this effect starts to appear even at the short block lengths used in this example. More research is required to understand this effect at moderate block lengths and to verify the same effect for LP decoding.

VI. CONCLUSIONS

In this paper, we present a novel linear-programing (LP) formulation of joint decoding for LDPC codes on FSCs that offer decoding performance improvements over joint iterative decoding. Joint-decoding pseudo-codewords (JD-PCWs) are also defined and the decoder error rate is upper bounded by a union bound sum over JD-PCWs. Finally, we propose

a simulation-based semi-analytic method for estimating the error rate of LDPC codes on FSISIC at high SNR using only simulations at low SNR.

REFERENCES

- [1] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linköping University, S-581 83 Linköping, Sweden, 1996.
- [2] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, M.I.T., Cambridge, MA, 2003.
- [3] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954–972, March 2005.
- [4] [Online]. Available: <http://www.PseudoCodewords.info>
- [5] P. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," 2007, accepted for *IEEE Trans. on Inform. Theory*.
- [6] R. R. Müller and W. H. Gerstacker, "On the capacity loss due to separation of detection and decoding," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1769–1778, Aug. 2004.
- [7] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *Eur. Trans. Telecom.*, vol. 6, no. 5, pp. 507–511, Sept. – Oct. 1995.
- [8] T. Wadayama, "Interior point decoding for linear vector channels," *Arxiv preprint cs.IT/0705.3990*, 2007.
- [9] M. F. Flanagan, "Linear-Programming Receivers," *Proc. 47th Annual Allerton Conf. on Commun., Control, and Comp.*, Sep. 2008.
- [10] M. F. Flanagan, "A Unified Framework for Linear-Programming Based Communication Receivers," *Arxiv preprint cs.IT/0902.0892*, 2009.
- [11] M. F. Flanagan, V. Skachek, E. Byrne, and M. Greferath, "Linear-programming decoding of nonbinary linear codes," *IEEE Trans. Inform. Theory*, vol. 55, no. 9, pp. 4134–4154, 2009.
- [12] M. H. Taghavi and P. H. Siegel, "Graph-based decoding in the presence of ISI," July 2007, submitted to *IEEE Trans. on Inform. Theory*.
- [13] C. Di, D. Proietti, E. Telatar, T. J. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.
- [14] T. Richardson, "Error floors of LDPC codes," *Proc. 42nd Annual Allerton Conf. on Commun., Control, and Comp.*, Oct. 2003.
- [15] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznik, "On the effective weights of pseudocodewords for codes defined on graphs with cycles," in *Codes, systems and graphical models*, ser. IMA Volumes Series, B. Marcus and J. Rosenthal, Eds., vol. 123. New York: Springer, 2001, pp. 101–112.
- [16] A. Kavčić, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1636–1652, July 2003.